Linear and Logistic Regression Workshop

Yong Hao and Philip

Objective

1. Introduction to basic machine learning models:
    - Linear Regression
    - Feature Engineering
    - Logistic Regression

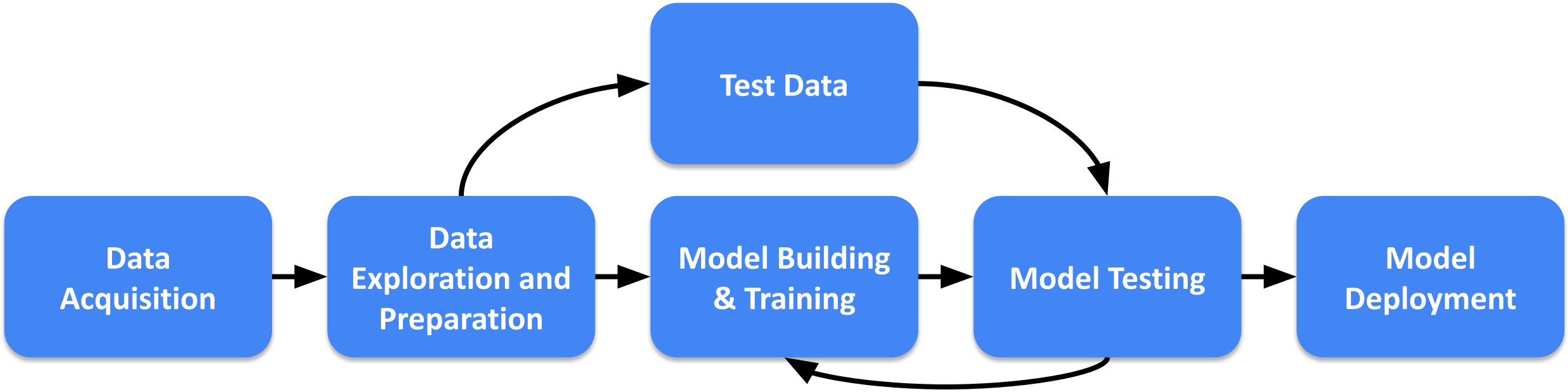
2. Problem Solving with Predictor Variables

3. Establish Foundation for Future Workshops

Supervised Learning VS Unsupervised Learning

Parameters	Supervised machine learning technique	Unsupervised machine learning technique
Process	Input and output variables given	Only input data given
Input Data	Algorithms trained using labelled data	Algorithms used against unlabelled data
Algorithms Used	Support vector machine, neural network, linear and logistic regression, random forest, classification trees	Cluster algorithms, K-means, hierarchical clustering
Computational Complexity	Simpler	Computationally complex
Accuracy of Results	Highly accurate and trustworthy method	Less accurate and trustworthy method
Real-Time Learning	Learning takes place offline	Learning takes place in real time
Number of Classes	Known	Not known

Supervised learning main pipeline

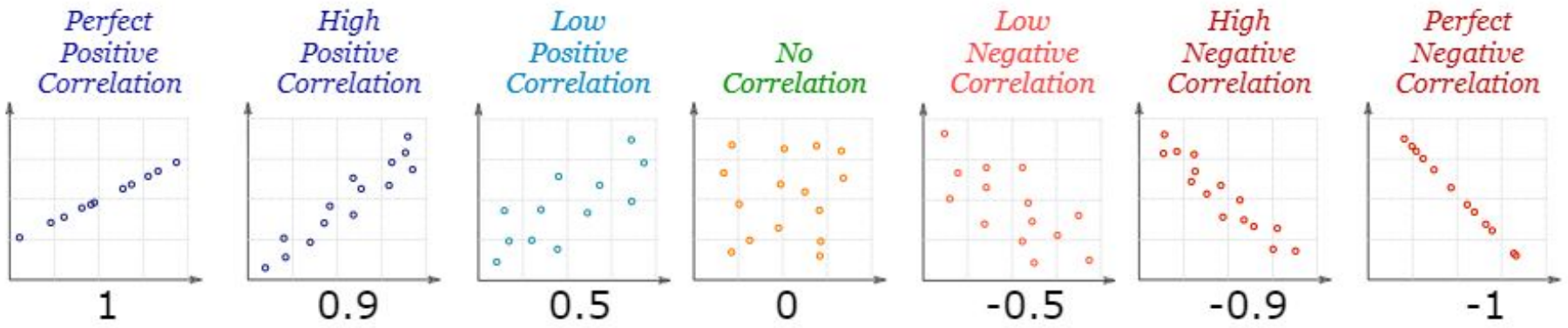


Linear Regression

Correlation

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

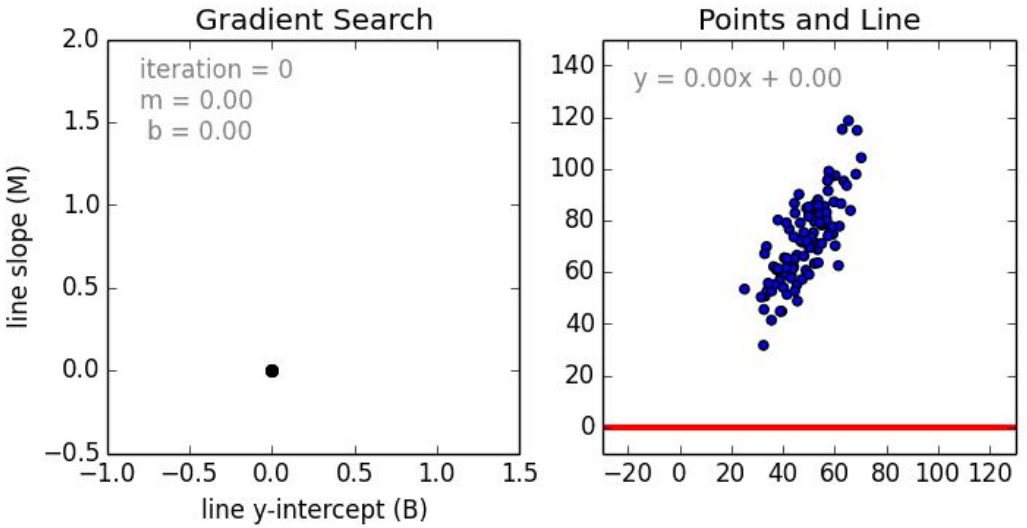
r = Pearson's correlation coefficient
x_i = values of the x variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable



Source: https://www.mathsisfun.com/definitions/correlation.html

Linear Regression

- ❑ Data is modelled with a straight line
- ❑ Takes continuous data as input and generates the corresponding output which is also continuous data



Source: https://techburst.io/introduction-to-linear-regression-bd7f834d0255

Applications:

- Predictions:
  - the outcome of political elections
  - the behavior of the stock market
  - the performance of a professional athlete
- Correlation:
  - effects of a proposed drug on the patients in a controlled study

Simple Linear Regression

- A statistical method used to summarize and study relationships between two continuous (quantitative) variables:
- One variable, denoted *x*, is regarded as the **predictor**, **explanatory**, or **independent** variable.
- The other variable, denoted *y*, is regarded as the **response**, **outcome**, or **dependent** variable.
- "simple" as it only uses one predictor variable.

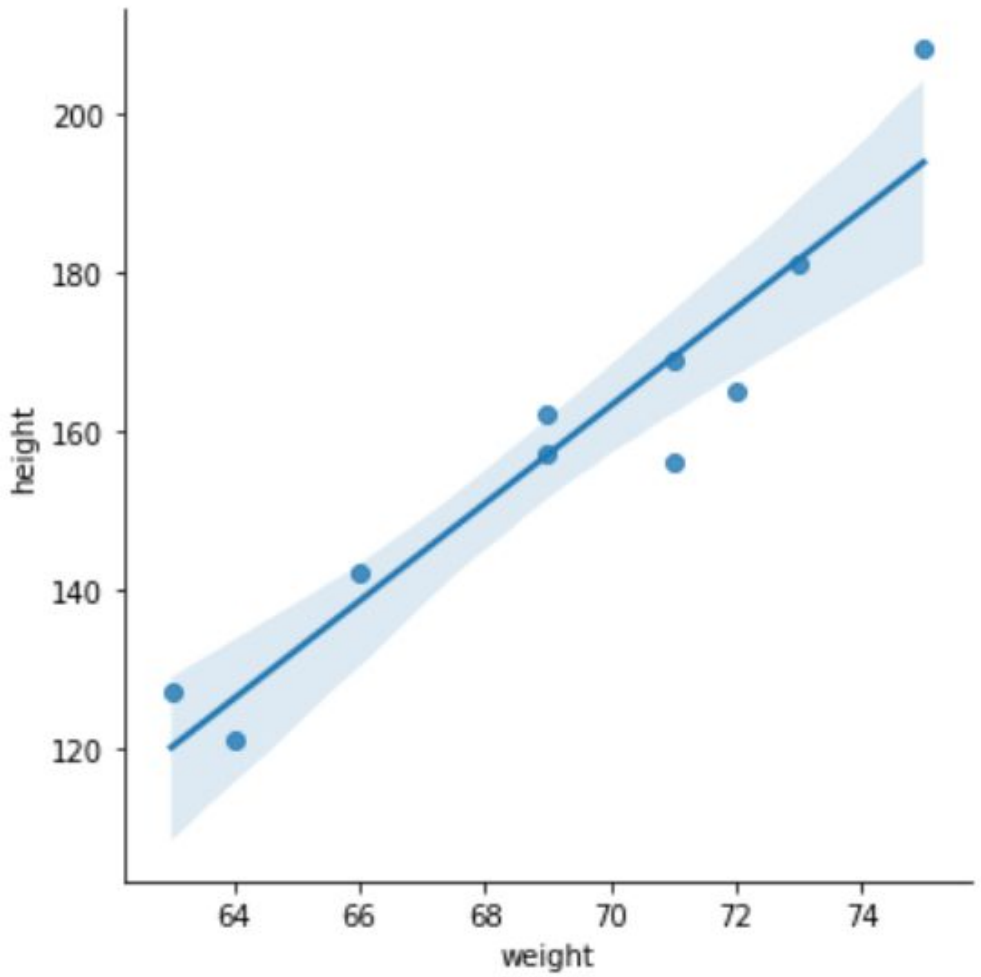
Weight, X (kg)	Height, Y (cm)
50	150
60	180
70	175

Pop Quiz

If we are trying to predict the sales of Chef Ramsay's steak based on weather, which variables are the predictor variable(s)?

A. Weather
B. Steak Sales
C. Weather and Steak Sales
D. Chef Ramsay's Hat

Simple Linear Regression



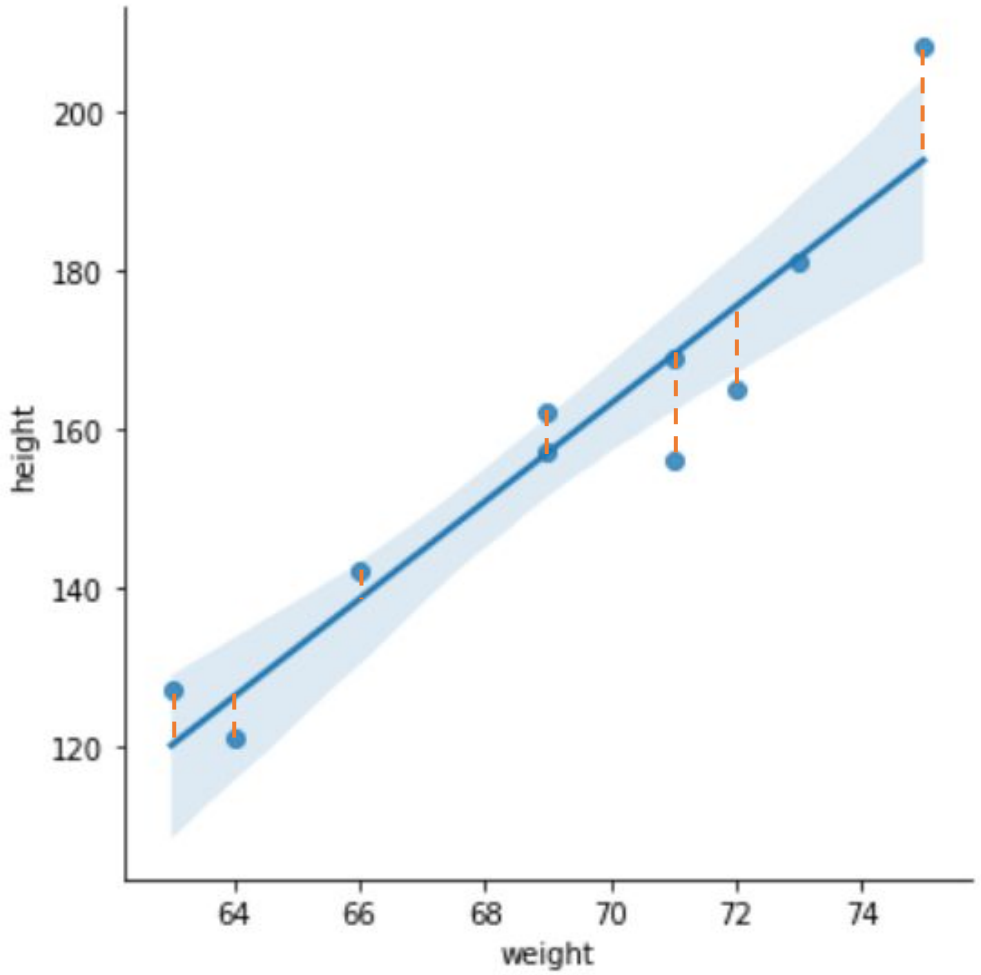
Simple Linear Regression Model:
$Y = \beta_0 + \beta_1 X_1 + \epsilon$

Y is the dependent variable
β_0 is the population Y-intercept
β_1 is the population slope coefficient
X_1 is the independent variable
ϵ is the random error term

Estimated Simple Linear Regression Equation:
$\hat{y} = b_0 + b_1 x_1$

x_1 is the input value
$\hat{y}$ is the predicted value
b_0 is the estimated Y-intercept
b_1 is the estimated slope coefficient

Simple Linear Regression



Predicton error for ith point

$$e_i = y_i - \hat{y}_i$$

Lost function: Squared error

$$Q = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Cost function: Loss function averaged over all training samples

$$Q = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Simple Linear Regression Formula

$$Q = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$$

Taking the derivative with respect to b_0 and b_1, set to 0, and solve for b_0 and b_1, we get the "**least squares estimates**" for b_0 and b_1:

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

for the **least squares regression line**.

The Normal Equation

$$\theta = \left(X^T X\right)^{-1} . \left(X^T y\right)$$

In the above equation,

θ : hypothesis parameters that define it the best.

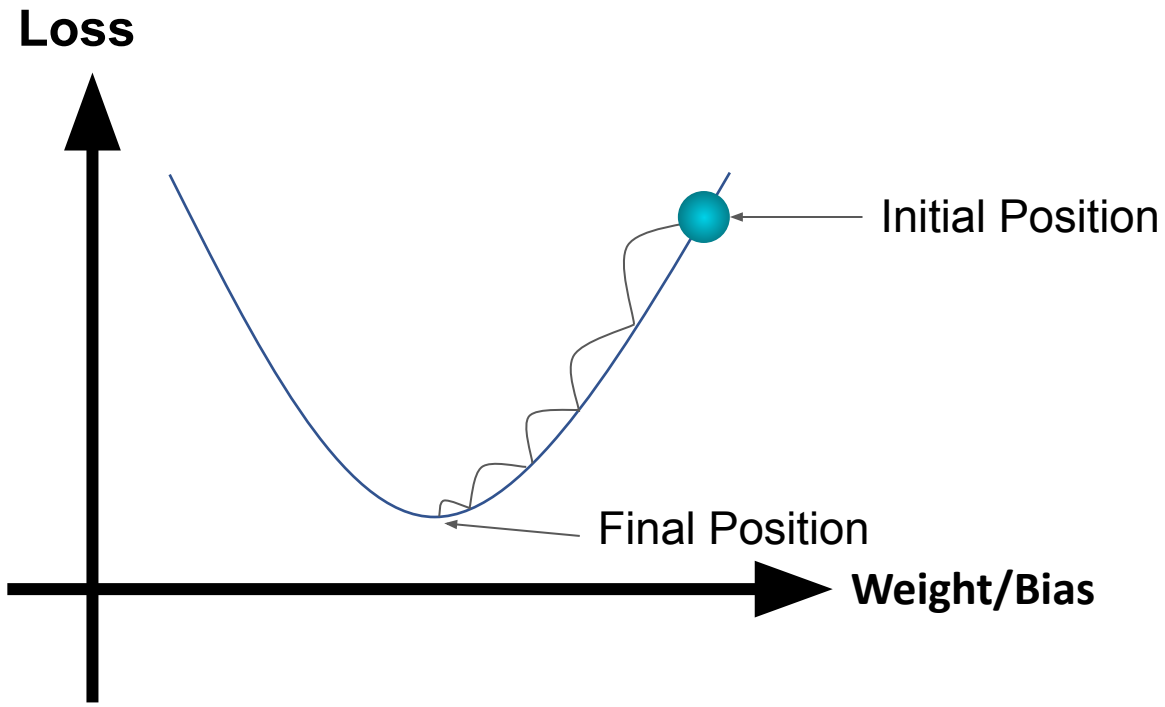
X : Input feature value of each instance.

Y : Output value of each instance.

- The more efficient way to implement the formula is by using matrices, bringing us to the normal equation
- An effective and a time-saving option when are working with a dataset with **small** features.
- It involves matrix inversion which is O(n^3) which is not scalable when we are dealing with **large** features

Gradient Descent Algorithm

- An iterative optimization algorithm to find the minimum of a function.
- It is O(n)



Gradient Descent Algorithm

Cost Function: $Q = \frac{1}{2n}\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2$

Taking partial derivative with respect to b_1:

$$D_{b_1} = \frac{1}{2n}\sum_{i=1}^{n} 2(y_i - (b_0 + b_1 x_i))(-x_i)$$

$$D_{b_1} = \frac{-1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i)$$

Taking partial derivative with respect to b_0:

$$D_{b_0} = \frac{-1}{n}\sum_{i=1}^{n}(y_i - \bar{y})$$

Let L be the learning rate which can be very small like 0.00001 for good accuracy.

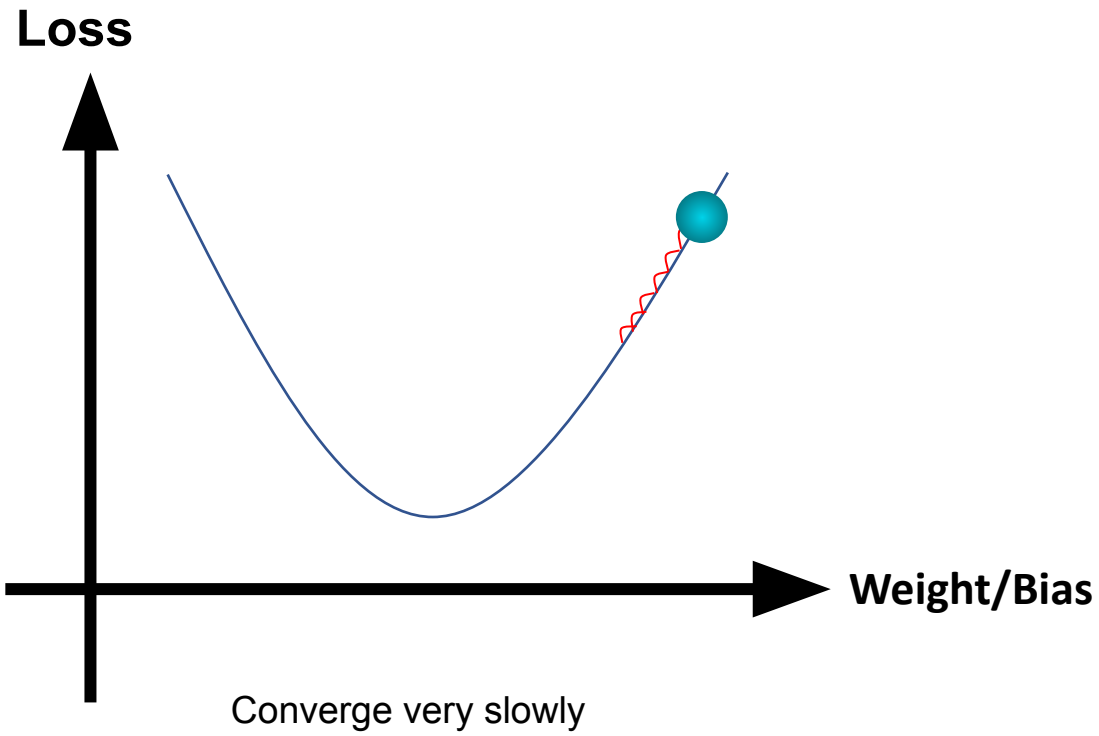
Updating the coefficients after every step:

$$b_1 = b_1 - L \times D_{b_1}$$
$$b_0 = b_0 - L \times D_{b_2}$$

Repeat these steps until the lost function returns 0 or after a set number of iterations

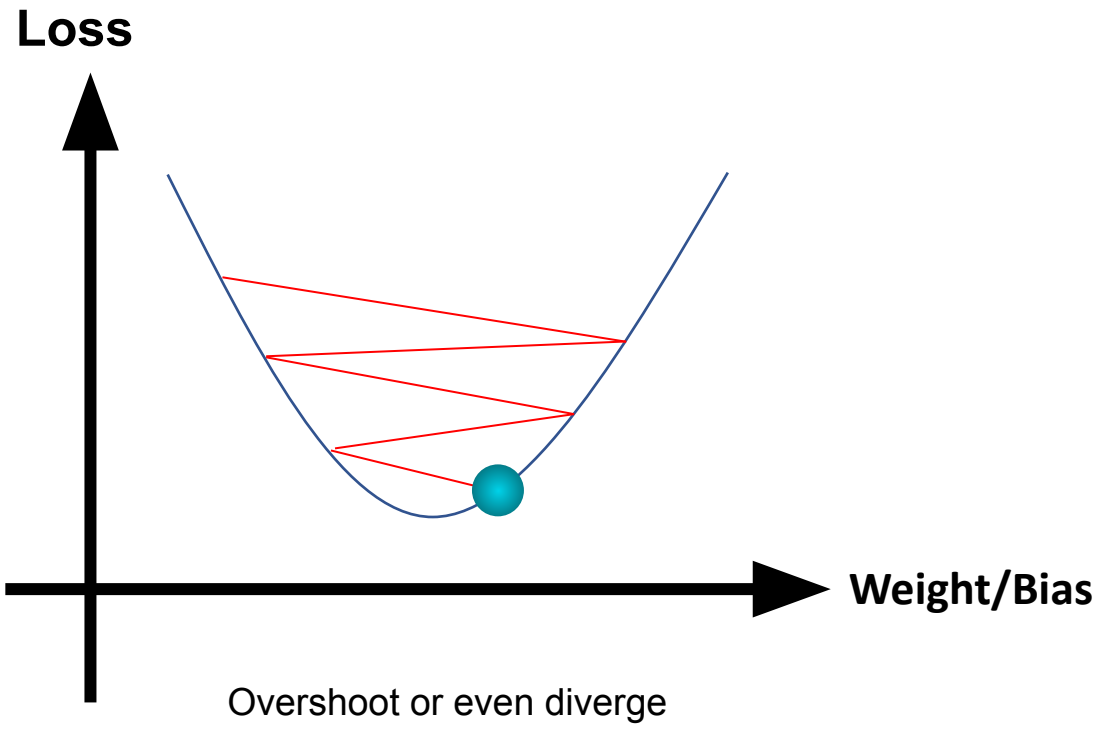
Learning Rate

Too Small



Converge very slowly

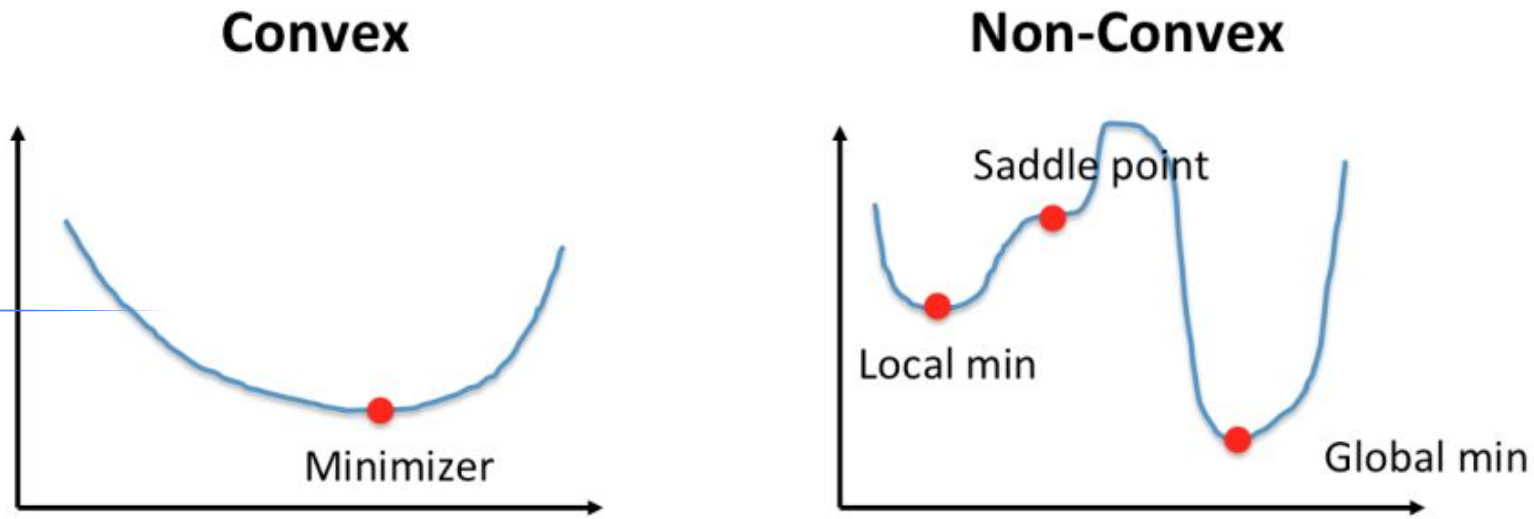
Too Big



Overshoot or even diverge

Convex and Non-convex Functions

Linear Regression
Logistic Regression



Source: https://pl.pinterest.com/pin/672232681861372201/

Features of a convex function

- Local minima = global minima
- The line segment between any 2 points on the graph lies above the graph and does not intersect the graph other than the 2 points

Gradient Descent (or any other optimization algorithm is guaranteed to find the minimum only for **convex functions** (if the learning rate is not too large and it went through enough iterations).

Pop Quiz

What is the cost function used for linear regression model optimization?

A. Mean Absolute Error
B. Sum of Errors
C. Mean Squared Error
D. Mean Error

Mean squared error is used to ensure that all errors are accounted for and penalise the larger errors

Multiple Variable Linear Regression

An extension of simple linear regression where 2 or more predictor variables are used to predict the variance of one response variable

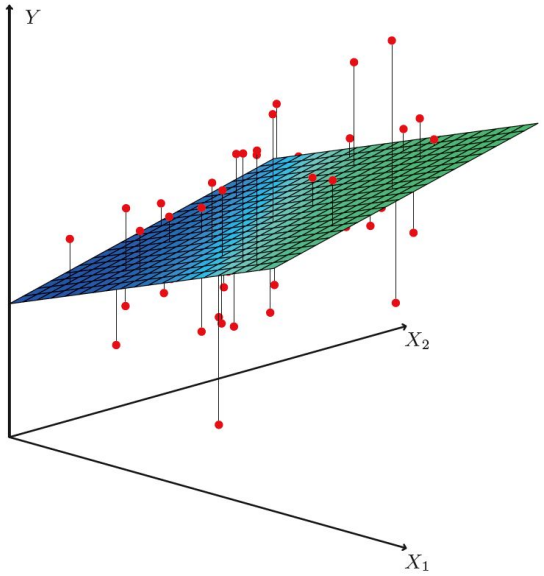
House Size, X_1	House Age, X_2	House Location, X_3	House Price, Y
$800m^3$	3 years	Bedok	$1,000,000
$200m^3$	8 years	Jurong East	$500,000

Multiple Linear Regression Model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Estimated Multiple Linear Regression Equation:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$$



Source: An Introduction to Statistical Learning: With Applications in R

Vectorization

Estimated Multiple Linear Regression Equation:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$$

$$\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$y(x) = \boldsymbol{b}^T \boldsymbol{x}$$

Why Vectorize?

- Using for-loops in Python is slow
- Vectorized code use highly optimised linear algebra libraries
- Matrix multiplication is very fast on Graphics Processing Unit (GPU)

Potential Issues With Multivariate Linear Regression

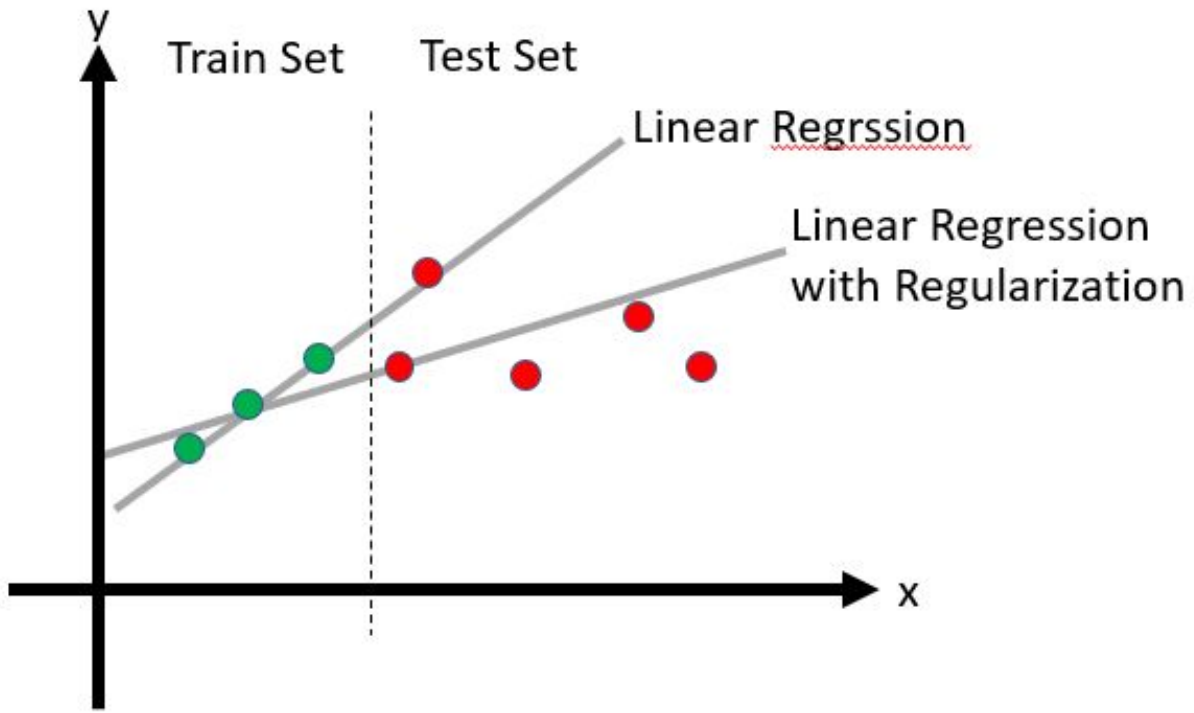
2 problems may arise:

- **Overfitting**: having too many predictor variables that account for more variance but do not add value to the model
- **Multicollinearity:** some predictor variables are correlated with each other and it becomes difficult to identify which variable is actually used in predictions

The ideal is for all predictor variables to be correlated with the response variables but not with each other

Regularization

A technique used to tune the model by adding an additional penalty term in the error function (reducing the weights and bias) to overcome overfitting, making the model less sensitive to variations in independent variables



Regularization

Linear Model: $\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$

Lasso Regression

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 + \lambda \sum_{j=1}^{p}|b_i|$$

- Uses absolute value as a penalty term
- Particularly useful for feature selection
- Can reduce slope to exactly zero

Ridge Regression

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 + \lambda \sum_{j=1}^{p} b_i^2$$

- Uses squared value as a penalty term
- Can reduce slope close to zero but not exactly zero

Evaluation Metrics for Linear Regression

- Mean Absolute Error (MAE)
  - $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
  - Easy to understand but does not punish large errors

- Mean Squared Error (MSE)
  - $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  - Larger errors are noted more than with MAE

Root Mean Squared Error

- $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- Has same units as y

R2 Score

- $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$
- "(total variance explained by model) / total variance."
- Expressed as a percentage

Evaluation of error should consider **context**!

Pop Quiz

What is regularization important in training your model?

A. Because the data has to be regular in size
B. Avoid overfitting
C. To make your model more lightweight
D. To add more terms in the lost function

Feature Engineering

Feature Engineering

What Creating features from raw data or adding functions of your existing features to your dataset

Why Allows us to reformulate non-linear problems as linear problems

How

→ Bivariate Combinations
→ Polynomials
→ Mathematical transformations
→ Dummy Variable Encoding

Example

Before $Y = \beta_0 + \beta_1(\text{height}) + \beta_2(\text{weight}) + \beta_3(\text{height}/\text{weight}^2) + \text{noise}$

After $Y = \beta_0 + \beta_1(\text{height}) + \beta_2(\text{weight}) + \beta_3(\text{BMI}) + \text{noise}$

Demo in Notebook

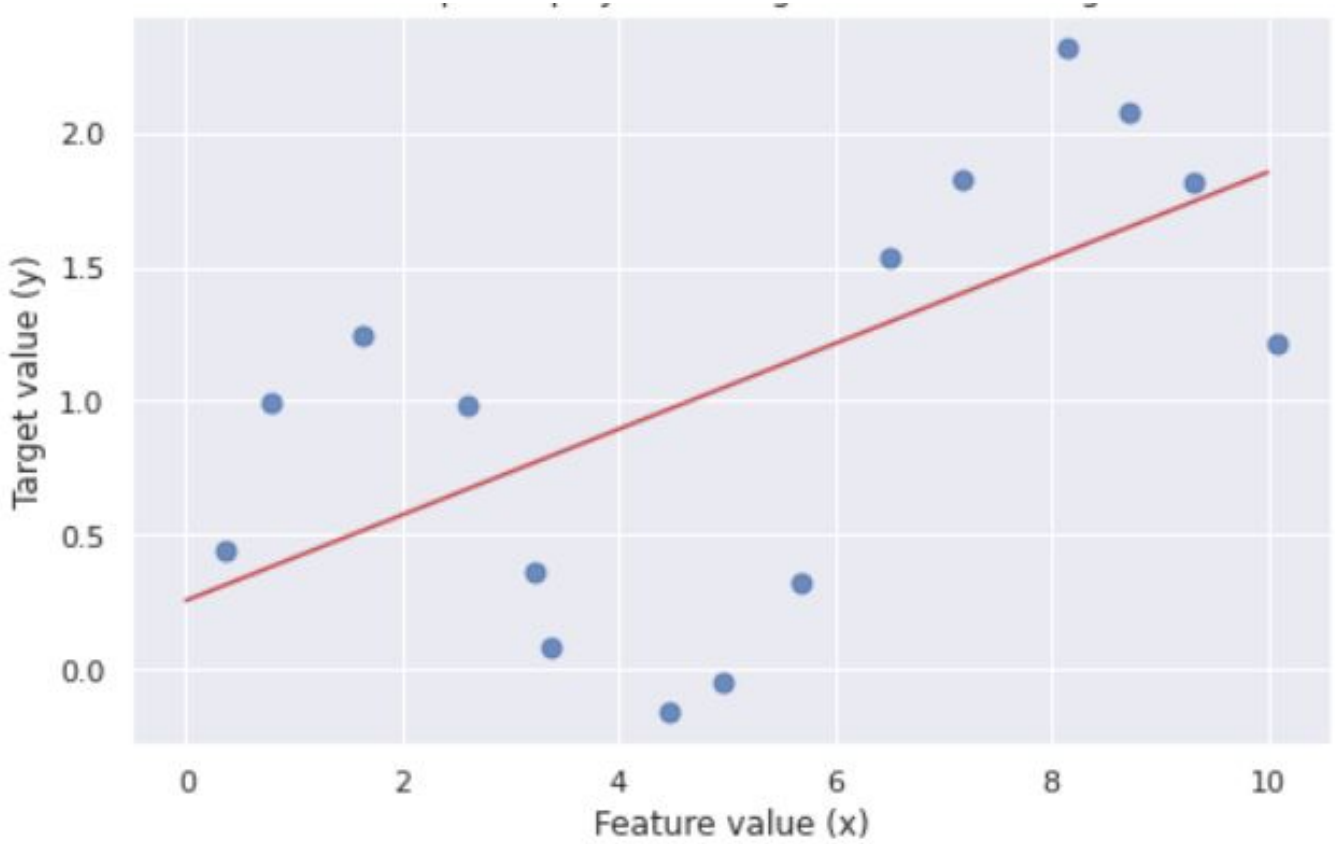
2 Examples:

- Log transform
- Dummy variable encoding

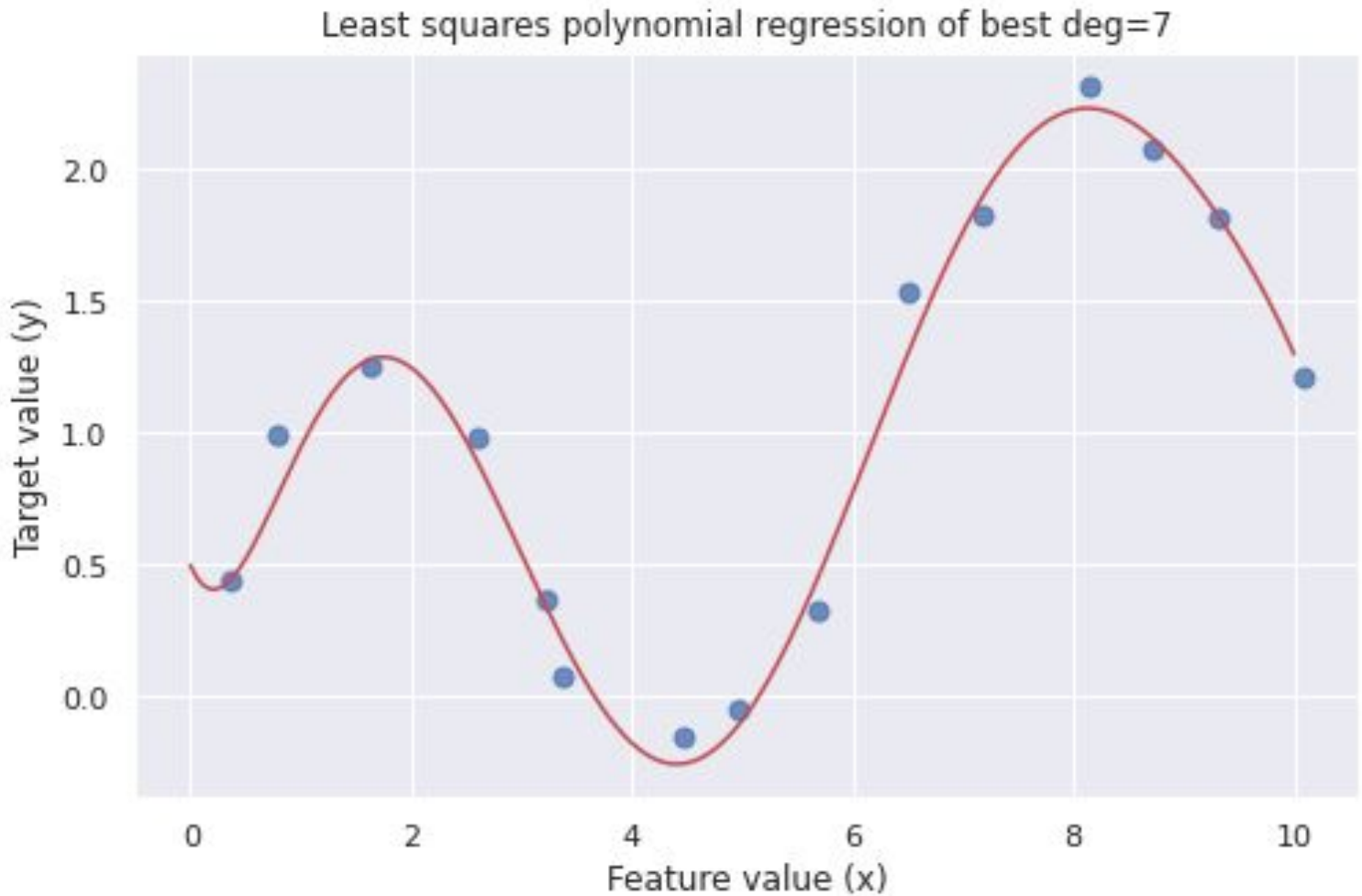
Polynomial Regression

What about this set of data?

In fact, R2-score < 0
does no follow trend
ie. worse than horizontal straight line



A better line of best fit



Polynomial Regression

- Linear features: y = mx + c = m(x) + c(1) -> Features: [1, x]
- Polynomial features: (Eg. 2nd degree)
  - We only have x0 x1 …
  - For x0: 1, x0, $x0^2$
  - For x0, x1: 1, x0, x1, x0x1, $x0^2$, $x1^2$

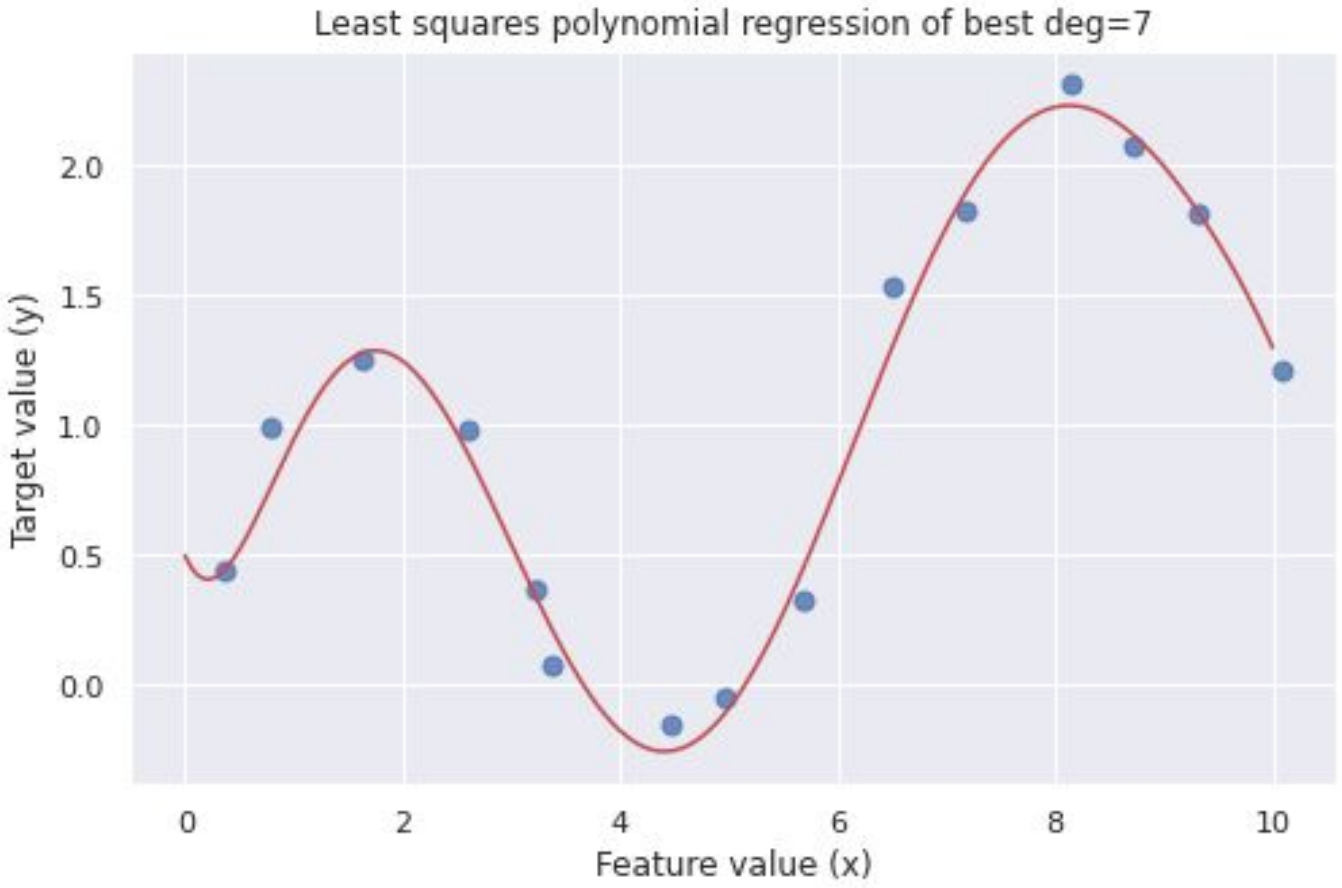
- Hence if its one-variable only, p(x) = a_0 + a_1*x + a_2*x^2 + ...
  - We add more columns of data

x	p(x)	x	x^2	x^3	...
1	-3	1	1	1	...
2	0	2	4	8	...

Demo!

Overfitting Note

What happens beyond the limits of the x i.e. new data?



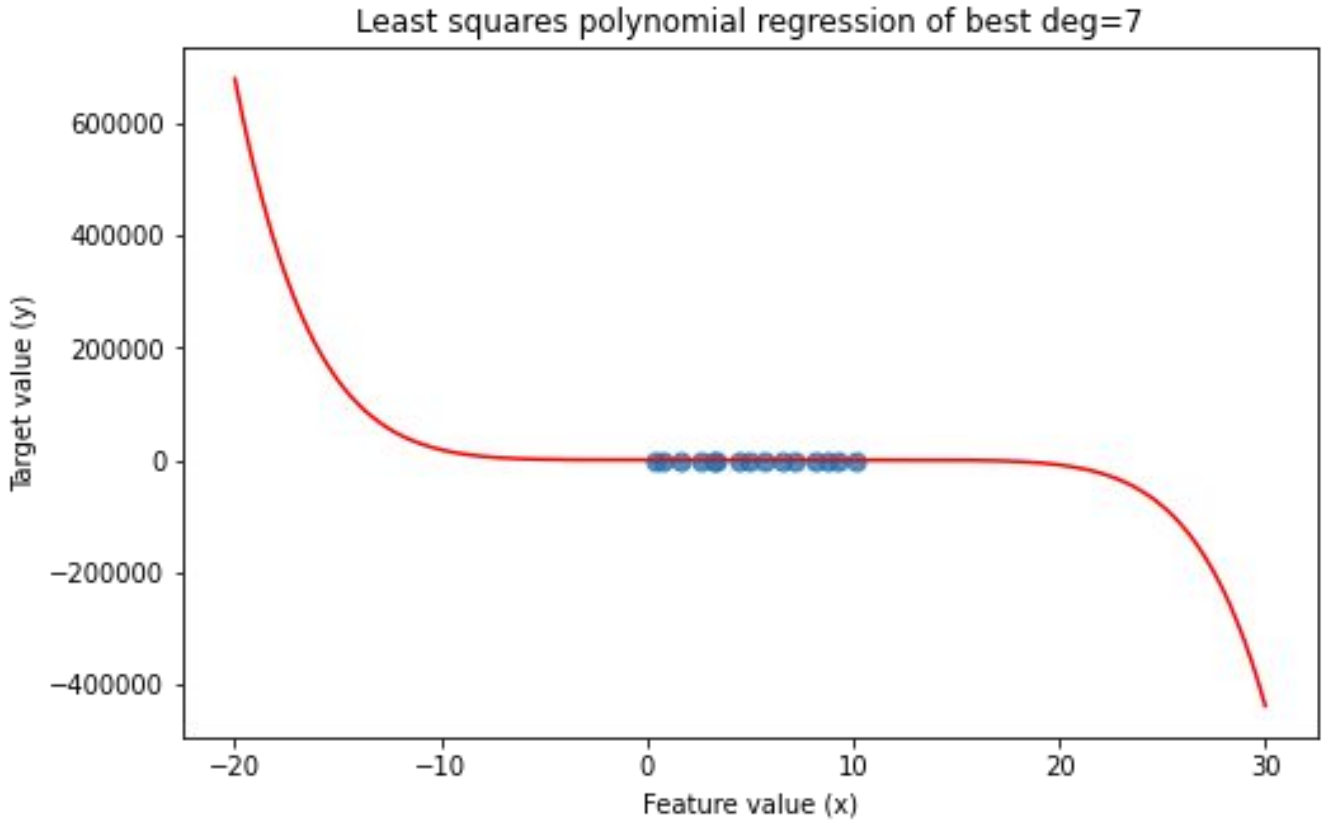
Overfitting Note

What happens beyond the limits of the x i.e. new data?

Might not be such a good idea to fit to the best curve …

Countermeasures learned in next few workshops

-> How to reduce overfitting



Pop Quiz

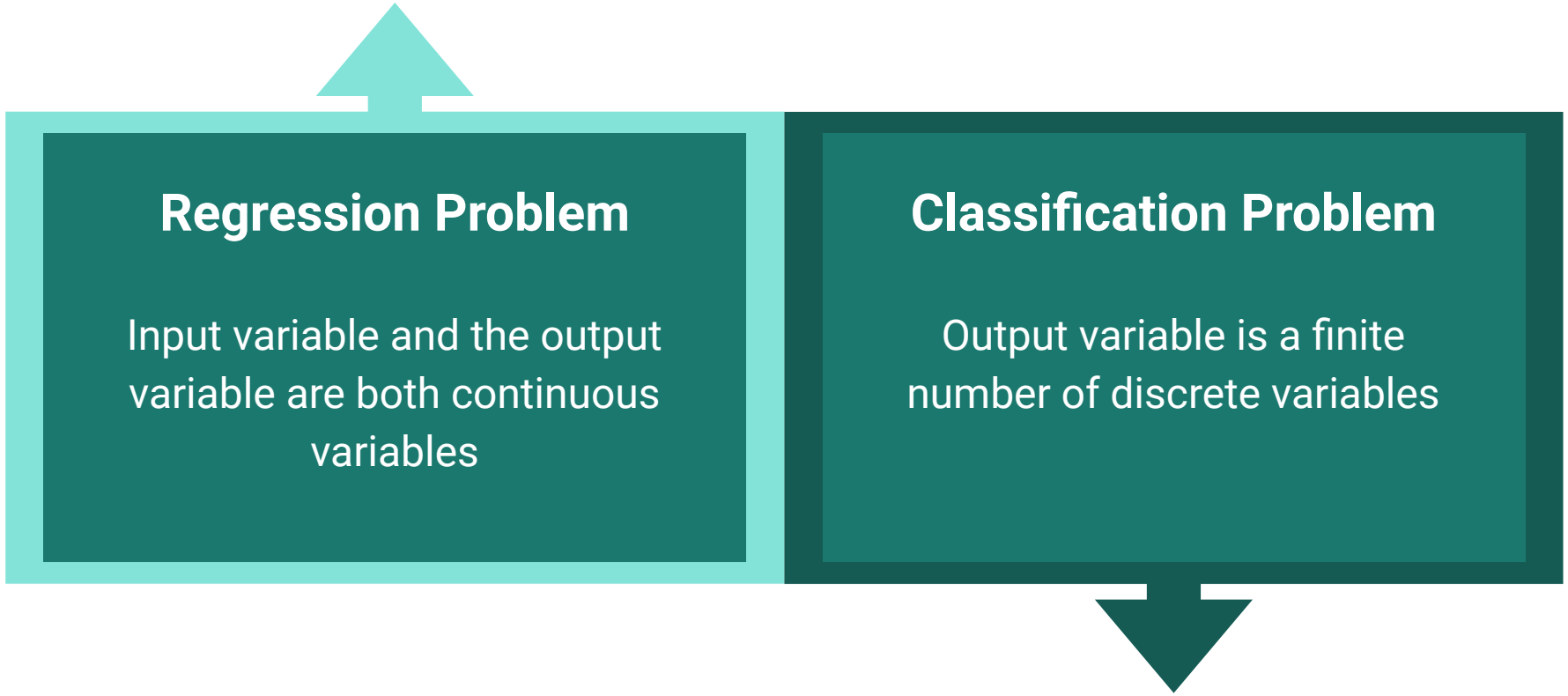
If a polynomial regression of degree 2 is fitted onto a dataset with 2 independent variables and 1 output, how many features will this model have?

A. 12 B. 6 C. 5 D. 2

- Polynomial features: (Eg. 2nd degree)
  - We only have x0 x1 …
  - For x0: 1, x0, $x0^2$
  - For x0, x1: 1, x0, x1, x0x1, $x0^2$, $x1^2$

Logistic Regression

Regression vs Classification



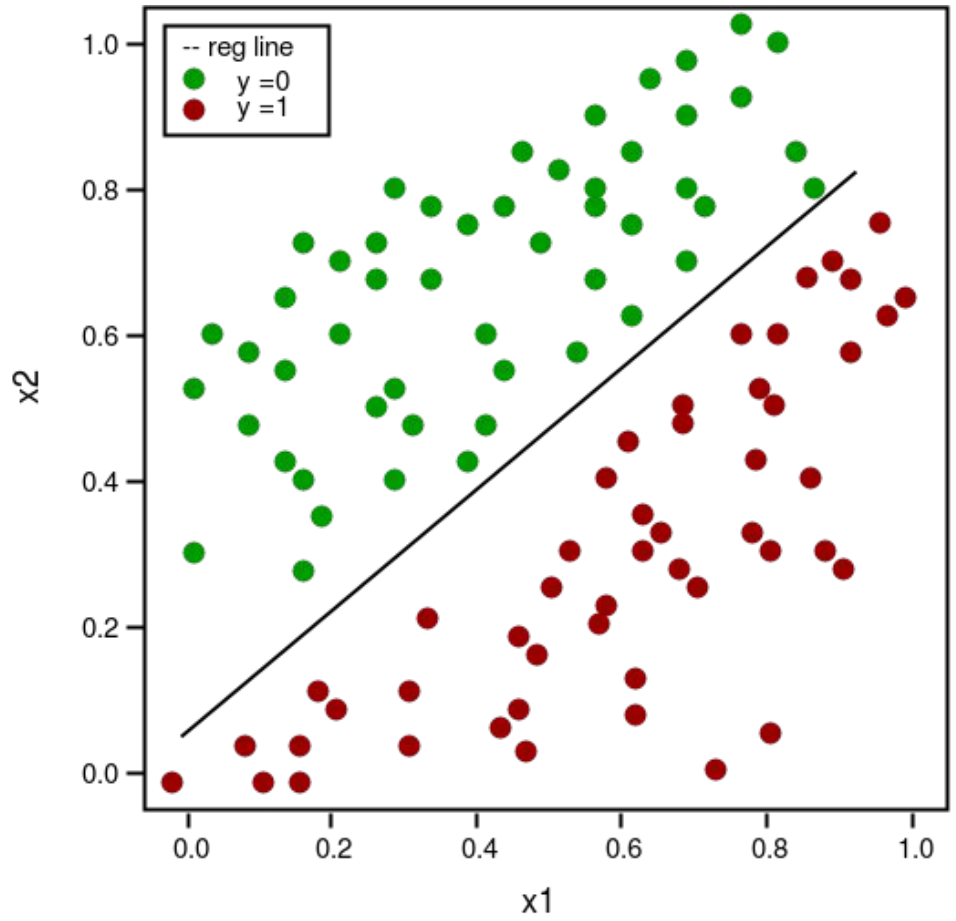
Logistic Regression

- A classification model, usually binary classification
- Loved for its simplicity, parallelization, and strong interpretability

Why is it called logistic regression then?

→ Underlying mechanism is still linear regression with 1 change

Decision Boundary



- Every point lying on the left/top of the line will be predicted as ‘1’
- ‘0’ for the other side of the line

How to find the decision boundary?

‘Curvy’ boundary
- needs feature engineering

Sigmoid/Logistic Function

$$y = \mathbf{w}^T\mathbf{x} + b$$

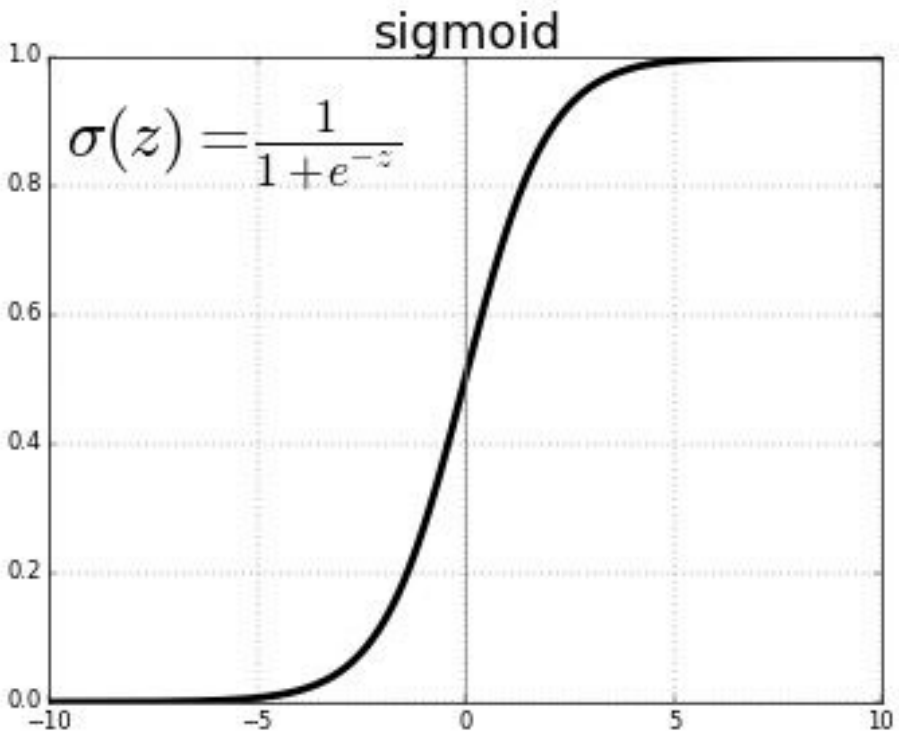
Bias (y-intercept)

Output

Weights (coefficients of x's)

Inputs

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

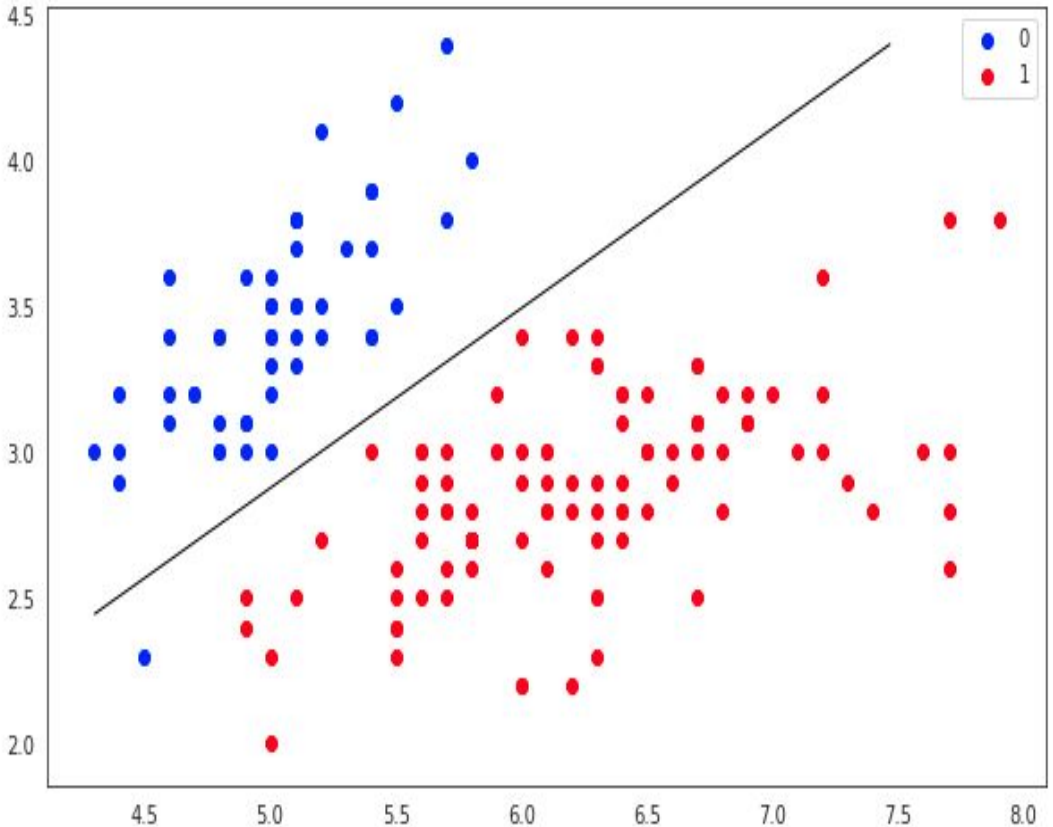


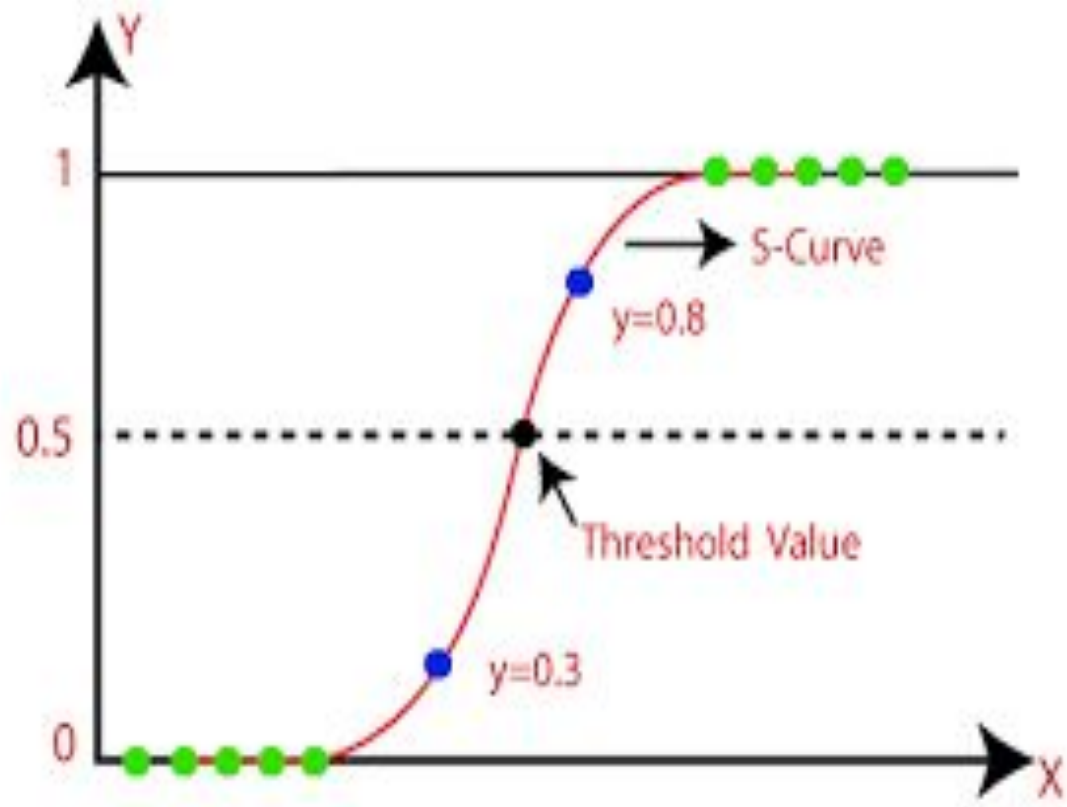
Also called ‘activation function’

w is a normal vector of the hyperplane that is the decision boundary

$$\theta_0 + \theta_1 x_1 + , \ldots , + \theta_n x_n = \sum_{i=1}^{n} \theta_i x_i = \theta^T x$$

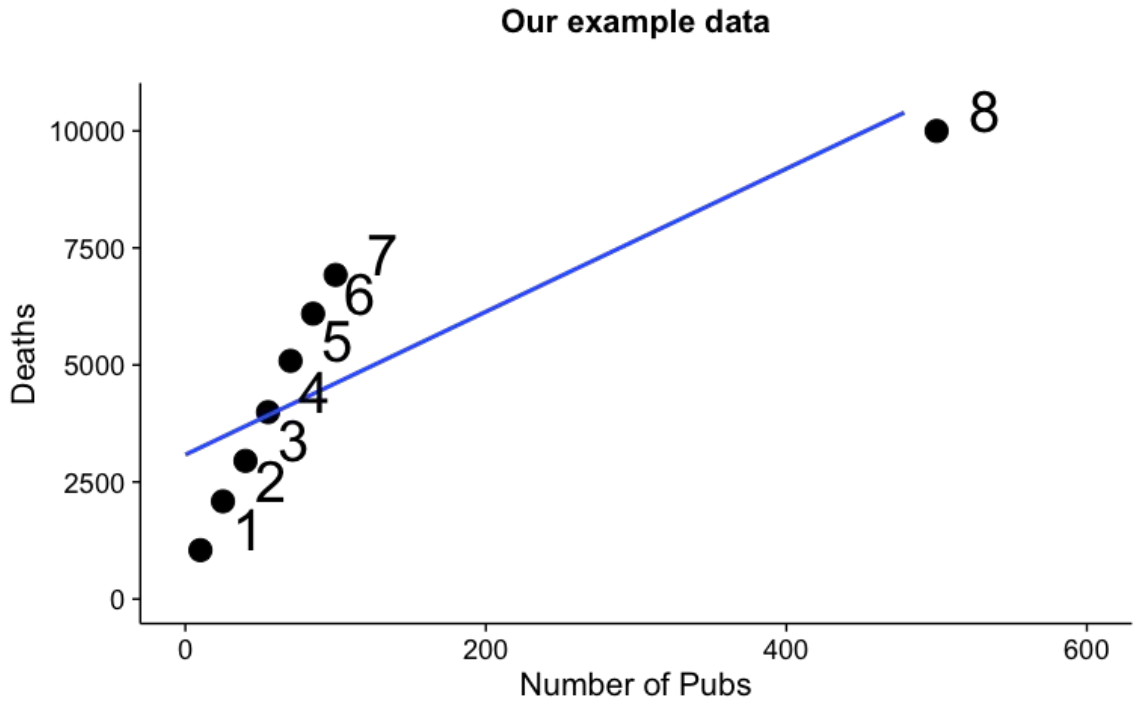
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$





Why not linear regression only?

1. Different problems - regression vs classification
2. Line of best fit is sensitive to outliers



Binary Classification

There is only two state of the outputs:

→ 1 or 0

The model regresses for the probabilities of a categorical outcome.

P(0)	P(1)	Pred
0.395	0.605	1
0.782	0.218	0
0.986	0.014	0
0.741	0.259	0
0.921	0.079	0
0.988	0.012	0
0.463	0.537	1
0.697	0.303	0
0.293	0.707	1
0.119	0.881	1

Pop Quiz

Why is the sigmoid function so important in logistic regression?

A. To engineer features of the model so that it will bounded between 0 to 1 for probability calculations
B. To squash the output of the model to [0, 1] so that outliers don't affect the decision boundary by a lot

An example to predict whether an outcome of '0' or '1' was used earlier. This is known as "binary classification". Can logistic regression be used then to predict more than two classes i.e. multiclass classification. (Yes/No)

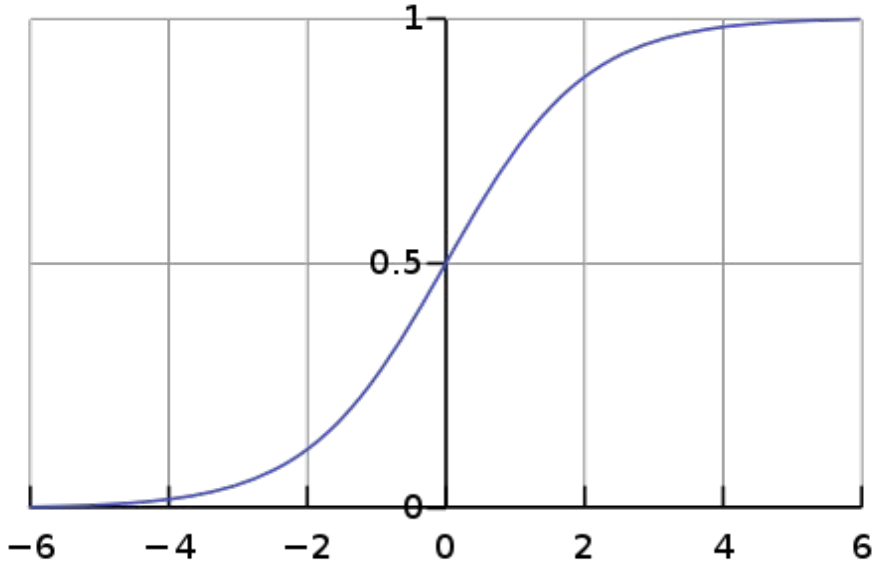
Logistic Regression Pipeline

1) Find the h function (hypothesis)
2) Find the loss function
3) Apply gradient descent method using the derivative of the loss function

Hypothesis

$$\theta_0 + \theta_1 x_1 + , \ldots , + \theta_n x_n = \sum_{i=1}^{n} \theta_i x_i = \theta^T x$$

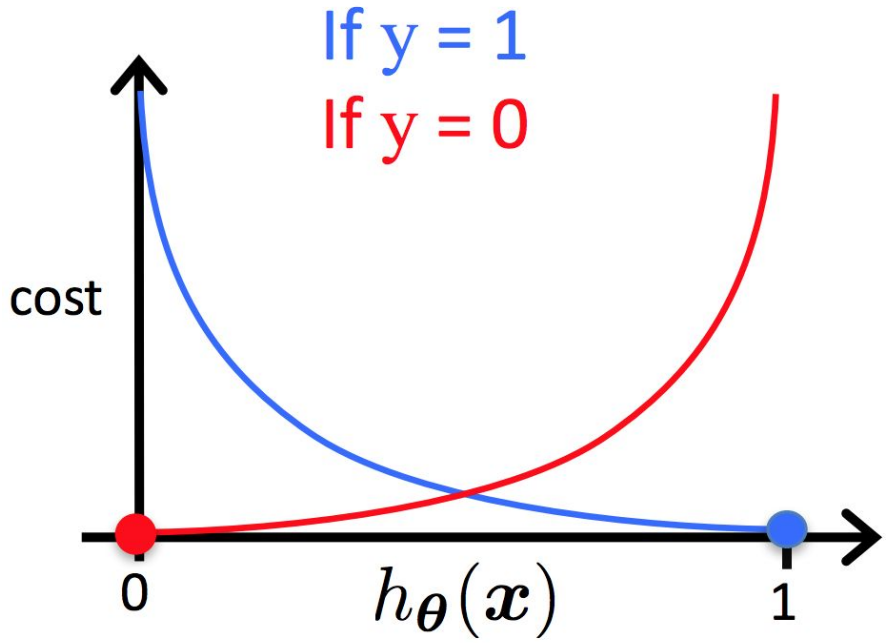
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



Loss function

The more ‘wrong’ the prediction, penalise it more heavily.

Just understand how …



$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{n} Cost(h_\theta(x^{(i)}), y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{n}\left(y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right)$$

m m

Apply gradient descent method

Update the weights using the gradient descent method with the formula below. Iterate till the loss converges to a minma

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

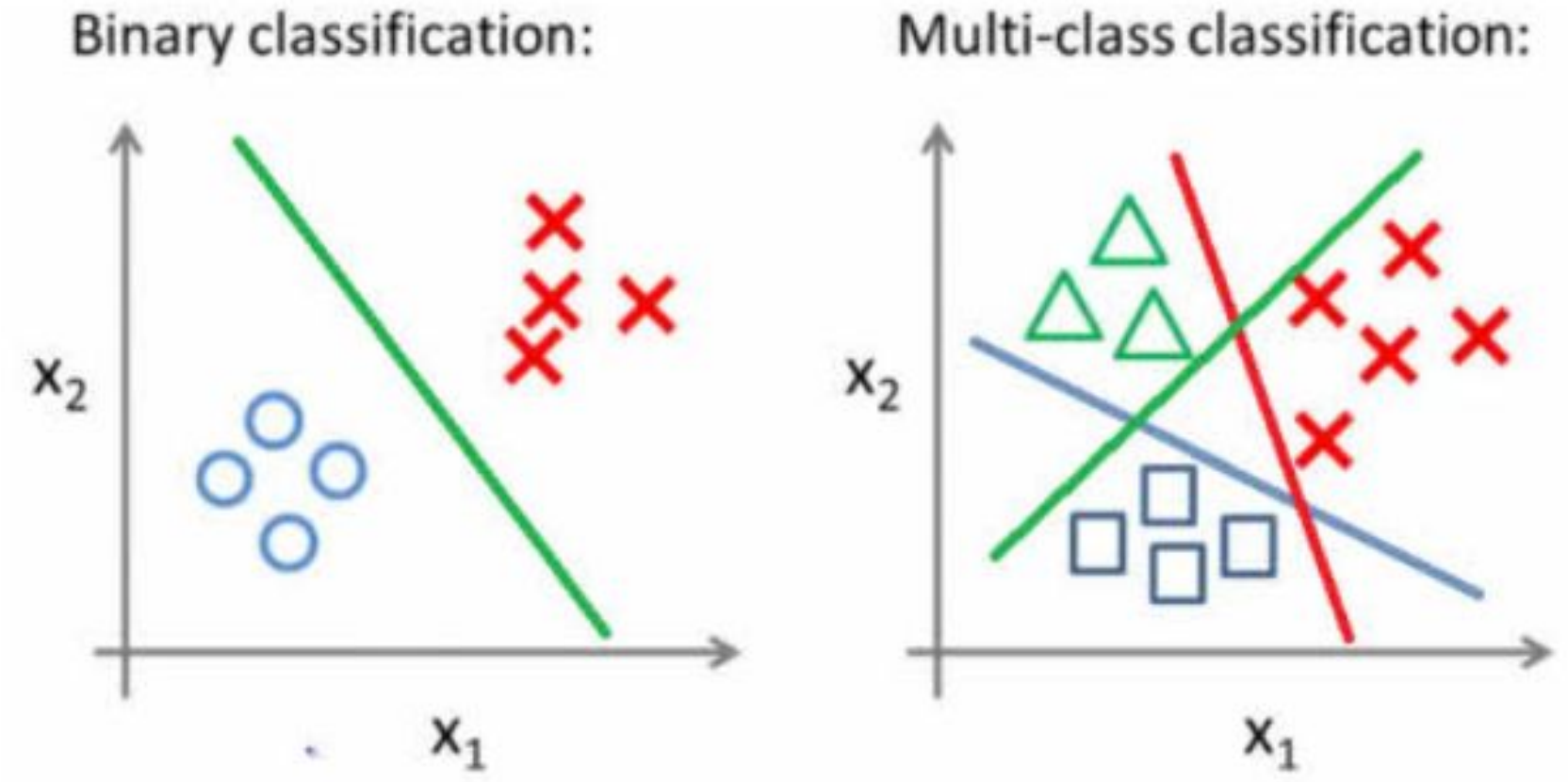
Pop Quiz

There is a false positive (0 label predicted as 1) and the h function outputs a number close to 0.9 for this data point.

How does this data point affect the overall cost?

A. It increases the cost by a lot.
B. It decreases the cost by a lot.
C. There is no change to the cost.

Multi-class Classification - Two Main Approaches

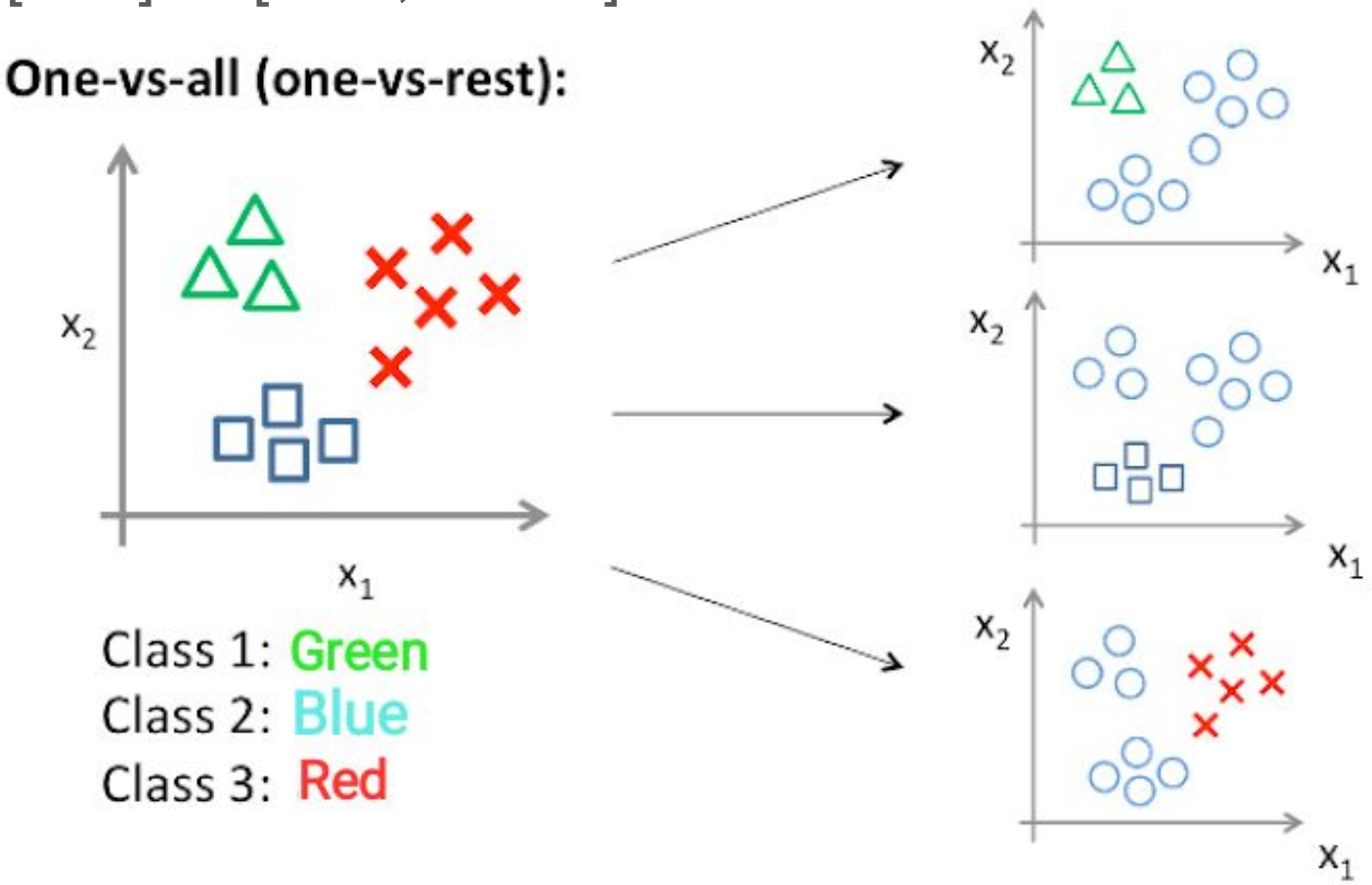


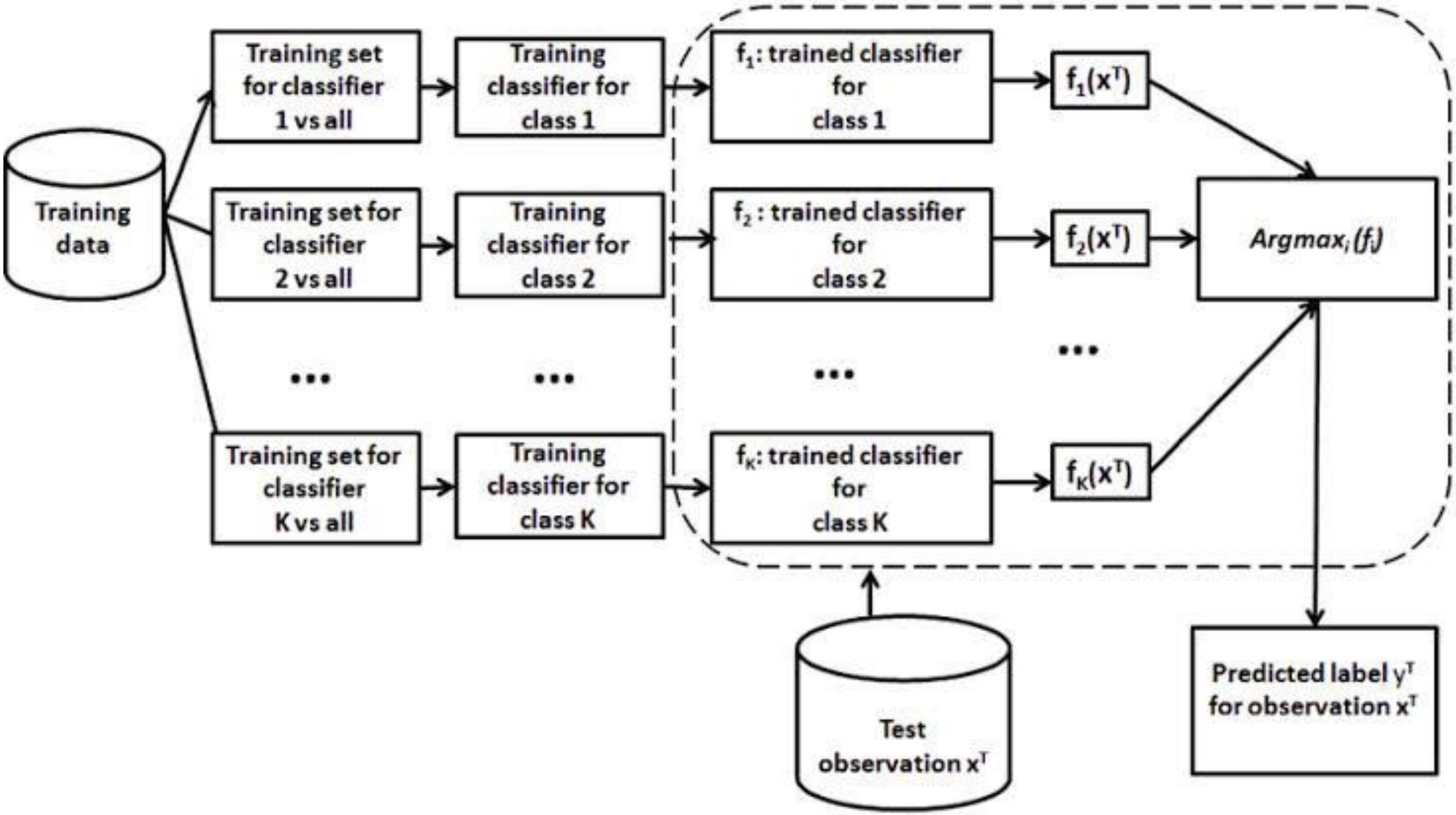
Multi-class Classification - One-vs-all

Classifier 1: [Green] vs [Red, Blue]

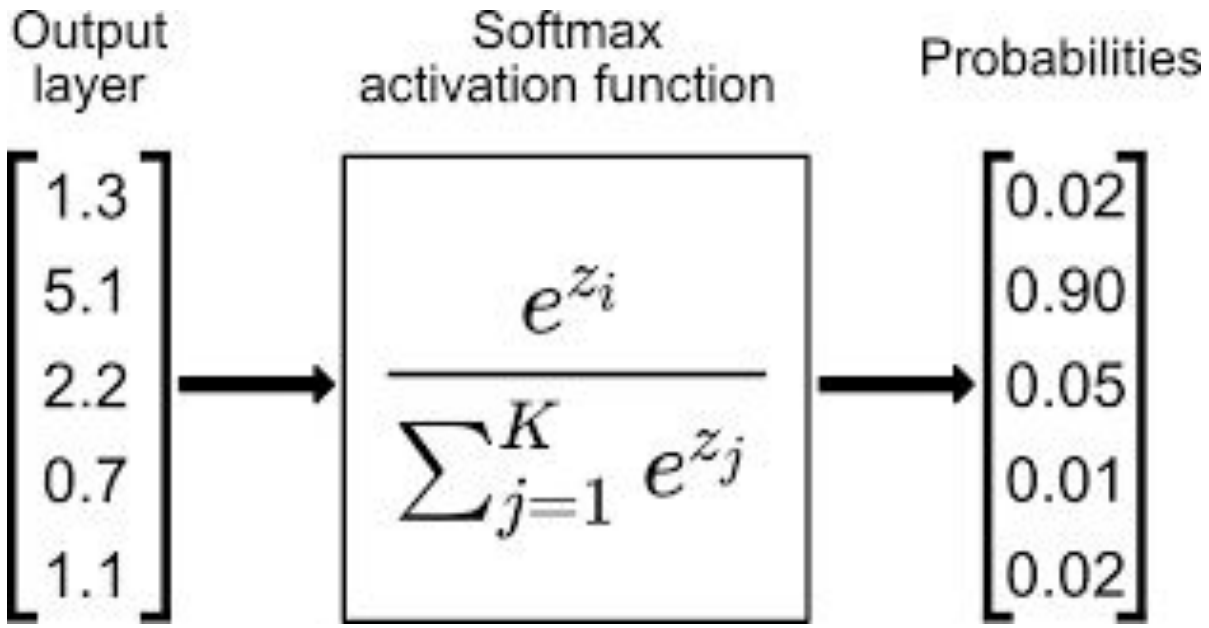
Classifier 2: [Blue] vs [Green, Red]

Classifier 3: [Red] vs [Blue, Green]




Training data
Training set for classifier 1 vs all
Training classifier for class 1
f1: trained classifier for class 1
f1(xT)
Training set for classifier 2 vs all
Training classifier for class 2
f2 : trained classifier for class 2
f2(xT)
Argmaxi (fi)
Training set for classifier K vs all
Training classifier for class K
fK: trained classifier for class K
fK(xT)
Test observation xT
Predicted label yT for observation xT

Multi-class Classification - Multinomial



$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Also used in neural networks!

Thank you! | Q & A

We Value Your Feedback!

https://forms.gle/Lkk76wWcDVJewgQK7